

**REMARKS****Status of the Claims**

Claims 1, 5-11, 15-21, and 25-30 are currently present in the Application, and claims 1, 11 and 21 are independent claims. Claims 1, 5, 7-11, 15, 17-21, 25, and 27-30 have been amended and claims 2-4, 12-14, and 22-24 have been canceled. Applicants are not conceding in this Application that those canceled claims are not patentable over the art cited by the Examiner, as the present claim amendments and cancellations are only for facilitating expeditious prosecution of the Application. Applicants respectfully reserve the right to pursue these and other claims in one or more continuation and/or divisional patent applications.

In particular, Applicants have amended independent claim 1 to include limitations similar to those previously found in dependent claims 2-4, and have therefore canceled claims 2-4. Similarly, Applicants have amended independent claims 11 and 21 to include limitations similar to those previously found in dependent claims 12-14 and 22-24, respectively, and have therefore canceled claims 12-14 and 22-24. Further support for the amendments to the claims is found in Applicants' specification on page 44, line 27 through page 55, line 5 (also see Figures 43-46). No new matter has been added as a result of the amendments to the claims.

Applicants appreciate the Examiner taking the time to discuss various issues pertaining to the Office Action via a telephone discussion on November 30, 2007. As a formal interview was not conducted, no agreement was reached regarding the claims during the telephone discussion.

**Claim Rejections - Alleged Anticipation Under 35 U.S.C. § 102 and Alleged Obviousness Under 35 U.S.C. § 103**

Claims 1, 8-11, 18-21, and 28-30 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Washington et al., U.S. Patent No. 5,835,775 (hereinafter Washington). Applicants respectfully traverse the rejections under 35 U.S.C. § 102.

Claims 2-7, 11-17, and 22-27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Washington. Applicants respectfully traverse the rejections under 35 U.S.C. § 103.

As noted above, Applicants have amended independent claim 1 to include limitations similar to those previously found in dependent claims 2-4, and have therefore canceled claims 2-4. Similarly, Applicants have amended independent claims 11 and 21 to include limitations similar to those previously found in dependent claims 12-14 and 22-24, respectively, and have therefore canceled claims 12-14 and 22-24.

Washington purports to teach a method and computer system for executing family generic, processor specific files (see Washington, Abstract). Applicants respectfully submit that Washington does not teach or suggest several of the elements of Applicants' independent claims. Using independent claim 1 as an exemplary claim, Applicants teach and claim the following:

- analyzing a source program for one or more program characteristics, the program characteristics selected from the group consisting of data locality, computational intensity, and data parallelism;
- in response to the analyzing, compiling the source program into two object files, a first object file corresponding to a first instruction set and a second object file corresponding to a second instruction set;
- in response to the compiling, storing the program characteristics in each of the object files;
- receiving a request to execute a software task corresponding to the source program;
- selecting a processor from a plurality of dislike processors, wherein a first processor corresponds to the first instruction set and a second processor corresponds to the second instruction set, to execute the software task, the selecting comprising comparing one or more characteristics of the software task with the program characteristics stored in the first object file and the second object file;

- in response to selecting the first processor:
  - loading the first object file into a shared memory, wherein the shared memory is shared by the plurality of dislike processors; and
  - executing the loaded first object file by the first processor; and
- in response to selecting the second processor:
  - loading the second object file into the shared memory; and
  - executing the loaded second object file by the second processor.

Applicants respectfully submit that Washington does not teach or suggest “analyzing a source program for one or more program characteristics, the program characteristics selected from the group consisting of data locality, computational intensity, and data parallelism,” as taught and claimed by Applicants. The Office Action cites Washington at col. 5, lines 50-52 as disclosing this element (see Office Action, page 8). However, the cited section of Washington discusses a first processor executing a processor test section which then “returns an identifier representing the processor type of the first processor.” Washington is not discussing analyzing a source program to determine characteristics of the program, as taught and claimed by Applicants. Rather, Washington discloses executing a program in order to determine what type of processor is being used to execute the program. Determining the type of processor currently being used to execute a program is not the same as analyzing a source program (i.e. a program that has not even been compiled, much less executed) in order to determine program characteristics, such as data locality, computational intensity, and data parallelism. Further, the type of program characteristics taught and claimed by Applicants, i.e. data locality, computational intensity, and data parallelism, are characteristics of the program itself, not characteristics of the processor on which the program will eventually execute.

Applicants also submit that Washington does not teach or suggest “compiling the source program into two object files, a first object file corresponding to a first instruction

set and a second object file corresponding to a second instruction set,” and then “storing the program characteristics in each of the object files,” as taught and claimed by Applicants. The Office Action notes that Washington does not disclose compiling a source program into two object files (see Office Action, page 7). However, the Office Action then states that “it would have been obvious to one having an ordinary skill in the art at the time the invention was made to recognize that compiling a source program into two object files and each adapted to be executed on a different processor is well known to the relevant art” (see Office Action, page 8). However, Applicants note that Washington discusses the use of separate object files (see Washington, Background of the Invention), but Washington specifically points out the drawbacks of using separate object files. Washington states that “[w]hat is needed therefore is a (sic) executable file that is optimized for a specific processor but executable by a family of processors” (Washington, col. 1, lines 61-63). The whole point of Washington is that Washington does not want to have separate object files for different processors, but rather, Washington’s goal is to produce one object/executable file that can be executed by an entire family of processors. Washington further states that “[i]t is yet another object of the present invention to provide a method for **generating a single program executable** that is optimized **for execution on at least two different processors**. It is yet a further object of the present invention to provide a method for **generating a single program executable** for a computer system having **two processors with different instruction sets**” (Washington, col. 3, lines 16-22, emphasis added). Therefore, Washington actually teaches away from Applicants’ claimed use of “a first object file corresponding to a first instruction set and a second object file corresponding to a second instruction set.”

Furthermore, Washington does not teach or suggest that “in response to the compiling, storing the program characteristics in each of the object files.” Applicants specifically claim analyzing a source program for program characteristics, compiling the source program into two object files, and then storing the program characteristics in each of these object files. As discussed above, Washington does not teach or suggest analyzing a source program for program characteristics. Further, Washington teaches

away from compiling the source program into two object files, each object file corresponding to a different instruction set. Therefore, because Washington does not disclose analyzing source program characteristics or compiling two object files, Applicants respectfully submit that Washington neither teaches nor suggests storing program characteristics in each of the object files, as taught and claimed by Applicants.

The Office Action cites Washington at col. 5, lines 58-59 as teaching storing program characteristics (see Office Action, page 8). However, the cited section of Washington discusses storing an identifier that indicates a processor type in a processor type flag. The cited section of Washington does not disclose storing characteristics of a source code program, such as such as data locality, computational intensity, and data parallelism, in each of two object files, as taught and claimed by Applicants.

Applicants further teach and claim “receiving a request to execute a software task corresponding to the source program,” and then “selecting a processor . . . to execute the software task.” Applicants claim that the processor is selected from a plurality of dislike processors and further claim a first processor corresponding to the first instruction set and a second processor corresponding to the second instruction set. Selecting a processor includes comparing characteristics of the software task with “the program characteristics stored in the first object file and the second object file.” Thus, if the first processor is selected based on the comparison, then the first object file is loaded into shared memory and executed by the first processor. Alternately, if the second processor is selected based on the comparison, then the second object file is loaded into shared memory and executed by the second processor.

Washington does not teach or suggest such selecting, comparing, loading, and executing. Rather, Washington does not appear to disclose any method for selecting a processor. It appears that processors are selected at random in Washington. Once a processor begins executing code, the processor first executes a processor test section that identifies the processor type of the processor (Washington, col. 5, lines 45-59). Based on the processor type, certain sections of the executable file will or will not be executed (Washington, col. 5, line 60 through col. 6, line 11). This is in direct contrast

to Applicants' claimed invention. Applicants specifically claim selecting a processor, from a plurality of dislike processors, based on a comparison of program characteristics (stored in the object files) and characteristics of the software task. Once a processor is selected, the applicable object code file is loaded and executed. Washington does not appear to perform any type of selecting based on comparing characteristics of a software tasks with program characteristics stored in object files. Further, Washington does not teach or suggest loading and executing one of the object files based on the selecting.

For the reasons set forth above, Applicants respectfully submit that Washington does not teach or suggest independent claim 1. Independent claims 11 and 21 include elements similar to those found in independent claim 1, and are patentable for at least the reasons discussed with regard to claim 1. Therefore, Applicants respectfully request that independent claims 1, 11, and 21, and the claims which depend from them, be allowed.

### **Conclusion**

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen, Reg. No. 42,196/  
Leslie A. Van Leeuwen, Reg. No. 42,196  
Van Leeuwen & Van Leeuwen  
Attorneys for Applicant  
Telephone: (512) 301-6738  
Facsimile: (512) 301-6742